CEWES MSRC/PET TR/98-34

WebHLA - An Interactive Programming and Training Environment for High Performance Modeling and Simulation

by

D. Bernholdt
G.C. Fox
W. Furmanski
B. Natarajan
H.T. Ozdemir
A. Odcikin Ozdemir
T. Pulikal

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC 94-96-C0002
Nichols Research Corporation

WebHLA - An Interactive Programming and Training Environment for High Performance Modeling and Simulation

D. Bernholdt, G. C. Fox, W. Furmanski, B. Natarajan, H. T. Ozdemir, Z. Odcikin Ozdemir and T. Pulikal

Northeast Parallel Architectures Center, Syracuse University

Presented at the DoD HPC 98 Users Group Conference, Rice University, Houston, TX, June 1-5 1998.

http://tapetus.npac.syr.edu/iwt98/pm/documents/dodugc98/paper.doc

1. Introduction

Our technology roadmap for High Performance Modeling and Simulation, outlined in NPAC PET FMS White Paper [1] is based on several ongoing and rapid technology evolution processes such as: a) transition of the DoD M&S standards from DIS to HLA; b) extension of Web technologies from passive information dissemination to interactive distributed computing based on enterprise standards offered by CORBA, Java and DCOM; c) transition of HPCC systems from custom (such as dedicated MPPs) to commodity base (such as NT clusters).

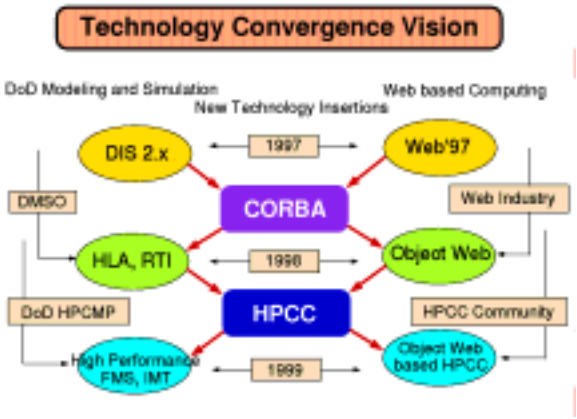


Figure 1: *Web, Enterprise, HPC and DoD M&S Technology Evolution Roadmap which underlies our WebHLA, approach.*

One common aspect of all these threads is the enforcement of reusability and shareability of products or components based on new technology standards. DMSO HLA makes the first major step in this direction by offering the interoperability framework between a broad spectrum of simulation paradigms, including both real-time and logical time models. However, HLA standard specification leaves several implementation decisions open and to be made by the application developers - this enables reusability and integrability of existing codes but often leaves developers of new simulations without enough guidance. In WebHLA, we fill this gap by using the emergent standards of Web based distributed computing, referred by some as the *Object Web* [2] that integrate Java, CORBA, W3C and DCOM models for distributed componentware. Traditional HPCC, dominated by data parallel MPP didn't make significant inroads into the DoD M&S where the focus is on task parallel heterogeneous distributed computing. Recent trends towards commodity based HPCC systems such as NT clusters offer a new promising framework for new generation high performance high fidelity M&S environments such as addressed by JSIMS, JWARS, JMASS or Wargame2000 programs.

We therefore believe that WebHLA, defined as the convergence point of the standardization processes outlined above will offer a powerful modeling and simulation framework, capable to address the new challenges of DoD computing in the areas of Simulation Based Design, Testing, Evaluation and Acquisition.

We are addressing WebHLA design and prototype development at NPAC in a set of PET FMS tasks at ARL and CEWES, including:

- Object Web based RTI which offers DMSO RTI (Run-Time Infrastructure) layer of HLA as a set of Java/CORBA objects, managed by JWORB - Java Web Object Request Broker under development at NPAC.
- Web based distance training for FMS systems and technologies such as core HLA, Object Web RTI, SPEEDES and later on also other FMS CHSSI systems under development.
- Designing Data Mining tools for the Virtual Proving Ground (VPG) project at ARL/ATC, Aberdeen, MD.

- Parallelizing CMS (Comprehensive Mine Simulator) developed at Ft. Belvoir, VA.
- Visual Authoring Tools for HLA simulations, using the WebFlow system at NPAC.

The overall architecture of our WebHLA prototype follows the 3-tier architecture of what we call *Pragmatic Object Web* [3] (see Fig. 2) with the RTI-over-JWORB based middleware, backend simulation modules (given by CMS, ModSAF etc. libraries, wrapped via CORBA/COM as FOM or SOM objects) and WebFlow based visual authoring front-ends.

In the following sections, we describe in more detail the WebHLA components listed above, followed by a summary and outlook towards WebHLA based Virtual Prototyping Environments for Simulation Based Design and Acquisition.

2. Object Web RTI

High performance simulation require integration of several M&S software modules, including simulators for tested objects, their parts, and suitable synthetic environments they operate in and interact with. Such integration of diverse simulation paradigms will be soon most conveniently performed within the new High Level Architecture (HLA) promoted by DMSO to enforce DoD-wide simulation interoperability. HLA includes the object models specified by the Object Model Template (OMT) and the Run-Time Infrastructure (RTI), acting as a software bus that supports interaction between HLA objects (federates) within applications (federations).

In parallel with these developments in the DoD M&S, new Web/Commodity standards are emerging or consolidating in the area of distributed objects and componentware such as CORBA, Java/RTI and DCOM. Current HLA is a custom distributed object model but DMSO's longer range plan includes transferring HLA to industry as CORBA Facility for Modeling and Simulation. Anticipating these developments, we are currently building within our HPCMP FMS PET activities at NPAC an Object Web based RTI prototype which builds on top of our new JWORB (Java Web Object Request Broker) middleware integration technology.

JWORB is a multi-protocol Java network server, currently integrating HTTP (Web) and IIOP (CORBA) and hence acting both as a Web server and a CORBA broker. Such server architecture enforces software economy and allows us to efficiently prototype new interactive Web standards such as XML, DOM or RDF in terms of an elegant programming model of Java, while being able to wrap and integrate multi-language legacy software within the solid software engineering framework of CORBA.

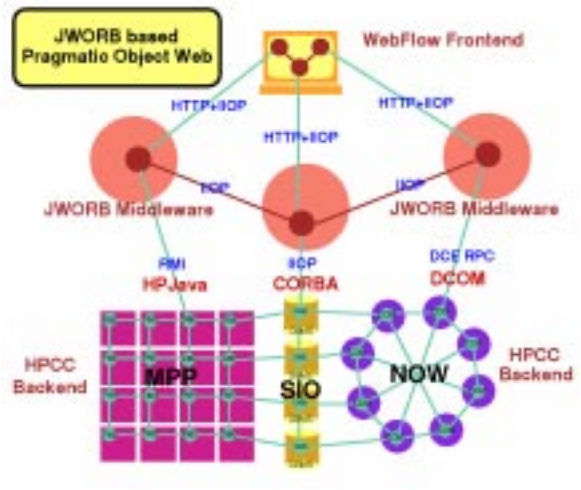


Figure 2: *Illustration of the communication protocol integration within our JWORB based Pragmatic Object Web. JWORB uses Java to integrate HTTP with IIOP and then it connects with NT clusters via COM/CORBA bridge.*

We are now testing this concept and extending JWORB functionality by building Java CORBA based RTI implementation structured as a JWORB service and referred to as *Object Web RTI* [4]. Our implementation includes two base user-level distributed objects: RTI Ambassador and Federate Ambassador, built on top of a set of system-level objects such as RTI Kernel, Federation Execution or Event Queues (including both time-stamp- and receive-order models). RTI Ambassador is further decomposed into a set of management objects, maintained by the Federation Execution object, and including: Object Management, Declaration Management, Ownership Management, Time Management and Data Distribution Management.

To be able to run C++ RTI demo examples, we developed a C++ library which: a) provides RTI C++ programming interface; and b) it is packaged as a CORBA C++ service and, as such, it can easily cross the language boundaries to access Java CORBA objects that comprise our Java RTI. Our C++ DMSO/CORBA glue library uses public domain OmniORB2.5 as a C++ Object Request Broker to connect RTI Kernel object running in Java based ORB. RTI Ambassador glue/proxy object forwards all method calls to its CORBA peer and Federate Ambassador, defined as another CORBA object running on the client side, forwards all received callbacks to its C++ peer.

3. Distance Training for the FMS Technologies and Systems

DoD M&S is a broad and rapidly evolving domain of distributed computing. Given the coming DoD-wide paradigm shift towards virtual prototyping, simulation based design and simulation based acquisition, we anticipate the growing demand on Web based distance training services in the M&S domain. DMSO is making the first step in this direction by publishing HLA specification, manuals and tutorials on the Web but more such services are needed, including interactive virtual programming laboratories in the M&S area.

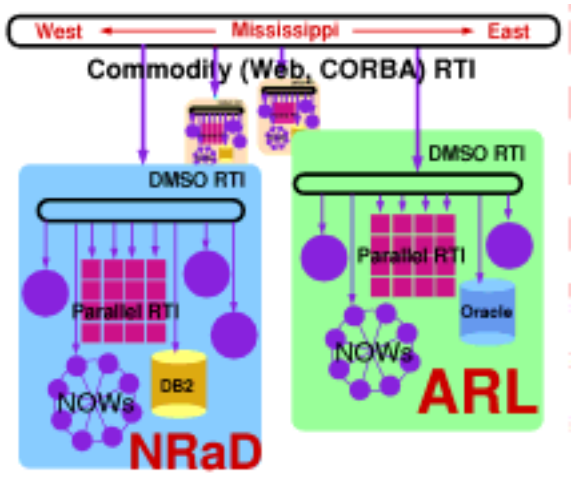


Figure 3: *Illustration of the interplay between DMSO RTI (running on Intranets), Parallel RTI (running on HPC facilities) and Commodity (such as Object Web) RTI. The latter is running in the Web / Internet Domain and connects geographically distributed MSRCs and DCs.*

We are developing and extensive electronic training space for FMS within the PET program, using our Object Web RTI as a core technology framework for distance training. In the HLA lingo, each participant of a training session becomes a federate, and so are their trainers/mentors as well as the particular M&S systems or technology modules selected as the current training target. The training session itself becomes a federation which follows the HLA rules for joining, participating, sharing the information, managing assets such as time, space or objects etc. Such training federations can be naturally made World-Wide distributed within our WebHLA framework (since Object Web RTI builds on top of JWORB which includes HTTP i.e. Web Server support) and given real-time interactive Web browser based interfaces (via ORBlets or equivalent ActiveX based dynamic connectivity channels to the CORBA/COM based training control components in the middleware).

Our current suite of FMS training materials in the development pipeline includes: a) HLA/RTI itself (including Web/distributed access to DMSO HLA demos); b) our Object Web RTI implementation; c) SPEEDES (Synchronous Parallel Environment for Emulation and Discrete-Event Simulation) training; d) CMS (Comprehensive Mine Simulator) training; e) ModSAF training (as required for CMS simulation). We are also interacting with FMS CHSSI projects and we plan to include other systems under development such as E-ModSAF, IMPORT, TEMPO or Thema into our evolving training suite.

4. Data Mining for the Virtual Proving Ground (VPG)

Our FMS user community includes currently the following groups: a) FMS CHSSI team where we provide PET training services for the FMS software (previous sections); b) Selected advanced M&S project teams such as the CMS team at Ft. Belvoir (next section); b) Virtual Proving Ground team at ARL/ATC in Aberdeen where we currently start providing help with the data mining services (this section) for the T&E historical test data.

In particular, we received recently from the VPG a MS Access database with a vehicle engineering and testing information and we analyzed it using some selected Web and data Mining technologies.

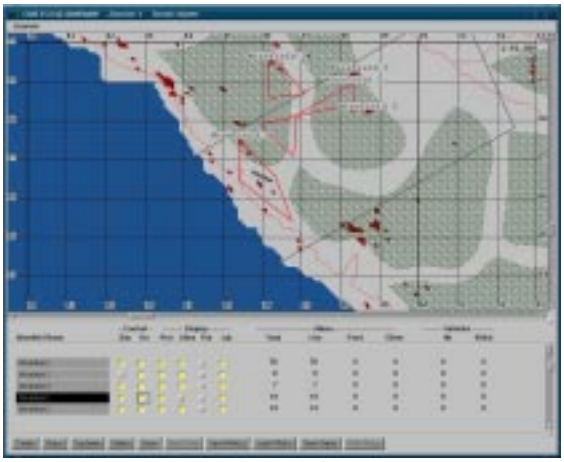
First, the VPG data was made accessible over the Web using Active Server pages for ease of access across workstations and network, and for use with future distributed datamining applications. An SQL Query tool was written on top of this data which can be used to run simple queries and analyze the results

In the second step, we decided to use simple classification algorithm, C4.5, for initial Data Mining experiments. We choose one of the attributes from the incident data table as our target class, dividing the values into two classes Major and Minor. We used a small subset of attributes that we thought would affect the classification process namely the subsystem, course condition and course type for our analysis. The public domain tool that we used was See5 from RuleQuest, which implements the C5.0 KDD algorithm which is a successor to Quinlan's C4.5 decision tree algorithm. Training data and test data were randomly selected with the help of the query tool. The See5 tool was used to run the algorithm over the training data to generate a decision tree and ruleset with an error rate of 3.8%. On the test cases the error rate was found to be 12%, which indicates the abnormalities in the training set selection and the decision tree generation. We are in

the process of refining this to get a lower error rate and to generate a better decision tree. We also calculated the information ratio for each of the selected attributes and the course condition was selected as the root node.

Data Mining experiments as described above allow us to become familiar with the large datasets and the high performance computational challenges of T&E. In the longer run, we view VPG as a promising testbed for WebHLA based virtual prototyping environments as envisioned in Section 7.

Figure 4: *Minefield Editor front-end within the Comprehensive Mine Simulator (CMS) environment from Ft. Belvior, VA.*



5. Parallelizing Comprehensive Mine Simulator (CMS)

CMS is a computationally demanding DIS system at Ft. Belvior, VA that simulates mines, mine fields, minefield components, standalone detection systems and countermine systems including ASTAMIDS, SMB and MMCM. The system can be viewed as a virtual T&E tool to facilitate R&D in the area of new countermine systems and detection technologies of relevance both for the Army and the Navy. We are currently analyzing the CMS source code and planning the parallel port of the system to Origin2000.

CMS simulates mine and other minefield objects, and their interactions with vehicles. Major CMS objects include: a) Mine (landmines including several conventional or custom types); b) Component (a collection of mines of one type), c) Minefield (a collections of components enclosed within a terrain perimeter). The CMS user interface supports a number of functions for editing mines during their creation, such as specifying their location or type.

Within CMS, a mine may interact with externally or internally simulated vehicles. Conceptually, the simulation objects operate largely independently. The control system allows these conceptually independent and autonomous objects to operate and interact without knowing whether other objects are on the same or other computer. In addition it controls the simulation timing. The control scheme uses messages to schedule and then execute processes that are conceptully independent. Due to well defined granularity and indepdendency of the simulation objects, parallel port of CMS to Origin2000 seems to be feasible even if the code is large and includes ModSAF libraries. We are currently identifying the critical loops over messages, mines, components and minefields that need to be optimized or/and paralleled and we intend to address the actual parallel port implementation in Year 3.

6. Visual Authoring Tools for HLA Simulations

DMSO has emphasized the need to develop automated tools with open architectures for creating, executing and maintaining HLA simulations and federations. The associated Federation Development Process (FEDEP) guidelines enforce interoperability in the tool space by standardizing a set of Data Interchange Formats (DIF) that are mandatory as input or output streams for the individual HLA tools. In consequence, one can envision a high-level user friendly e.g. visual dataflow authoring environment in which specialized tools can be easily assembled interactively in terms of computational graphs with atomic tool components as graph nodes and DIF-compliant communication channels as graph links.

Within our HPCMP FMS PET project at NPAC we are building such visual HLA tool assembly framework on top of the NPAC WebFlow [5] system. WebFlow is a Web/Java based visual dataflow environment with the Web browser based computational graph editor and the runtime given by a mesh of interconnected Java Web Servers, used to manage WebFlow middleware module wrappers and acting as proxies to the backend computational modules.

Through WebFlow linkage with HPC via modules/wrappers for Globus Metacomputing as well as support for database and visualization, our approach offers a natural platform for adressing HPC and HLA integration issues.

We started this project by analyzing currently existing tools and their limitations. In particular, we inspected the Object Model Development Tool (OMDT) by Aegis Research Center, Huntsville, AL as a representative current generation DMSO FEDEP

tool, and the OSim Tool by OriginalSim, Montreal, CA as a representative current commercial product in this area.
OMDT is a Windows 95/NT-based application that supports building HLA Object Model Template (OMT) tables such as Class, Interaction or Attribute Tables using a spreadsheet-like user interface OSim is a suite of commercial simulation development frameworks designed to facilitate the federate building process through automation and visual authoring.

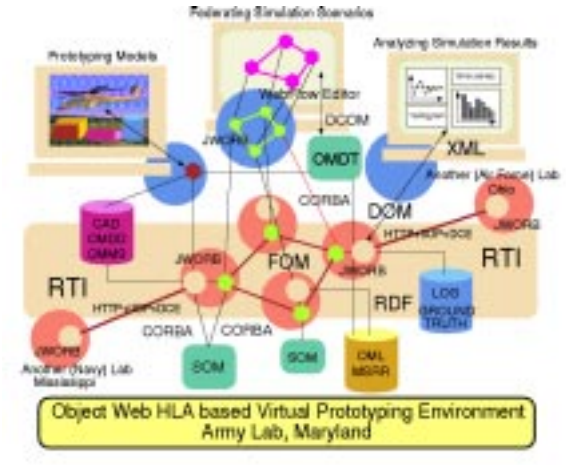


Figure 5: *An illustration of a Virtual Prototyping Environment for Simulation Based Acquisition – an eventual goal for the WebHLA approach presented in this document.*

We found that current tools, albeit useful in the standalone mode, are not yet fully ready to act as standardized reusable components in larger toolkits. For example, OMDT is a standalone MFC application accessible only via a human-driven GUI, not a COM server with automation/scripting support, whereas OSim's custom visual builder can be now more naturally addressed in terms of the UML (Uniform Modeling Language) extensible standard for visual modeling.

Based on lessons learned in the analysis of the current generation tools, we are in the process of building an OMDT-like editing tool based on Microsoft Component Object Model (COM) architecture. Rather than building our sample tool from scratch, we construct it by customizing Microsoft Excel Component using the Visual Basic for Applications and the OLE automation methodology. Using this approach, we were able to emulate the look-and-feel of the OMDT tool, while at the same time packaging our tool as a reusable COM or ActiveX component that can smoothly cooperate with other visual authoring tools within the WebFlow model.

7. Summary and Outlook

We have presented an overview of a set of ongoing FMS PET activities at NPAC that collectively develop WebHLA – an interactive programming and training environment for high performance modeling and simulation. We described first our JWORB based Object Web RTI which integrates Web, Enterprise and M&S protocols and offers a core framework for interactive hands-on FMS training. Next, we discussed a few specific high level tools such as Data Mining for VPG datasets or visual FOM/SOM authoring, and selected high performance M&S applications such as CMS.

Fig 5 illustrates our envisioned end product in the WebHLA realm – a distributed, Web / Commodity based, high performance and HLA compliant Virtual Prototyping Environment for Simulation Based Acquisition with Object Web RTI based software bus, integrating a spectrum of M&S tools and modules, wrapped as commodity (CORBA or COM) components and accessible via interactive Web browser front-ends. Such environments, currently operated only by large industry such as Bowing, become affordable within the current technology convergence process as envisioned in Fig. 1 and quantified in our WebHLA integration program.

8. References

[1] D. Bernholdt, W. Furmanski and G. C. Fox, Towards High Performance Object Web based FMS, White Paper for the FMS CTA, Sept 1997.

[2] Robert Orfali and Dan Harkey, Client/Server Programming with Java and CORBA , 2nd Edition, Wiley 1998.

[3] G. C. Fox, W. Furmanski and S. Pallickara, Building Distributed Systems for the Pragmatic Object Web, Wiley 1998 book in progress.

[4] G. C. Fox, W. Furmanski and H. T. Ozdemir, Java/CORBA based Real-Time Infrastructure to Integrate Event-Driven Simulations, Collaboration and Distributed Object/Componentware Computing, submitted to PDPTA'98, Las Vegas, July 98.

[5] D. Bhatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski and G. Premchandran, WebFlow - a visual programming paradigm for Web/Java based coarse grain distributed computing , June '97, in *Concurrency: Practice and Experience.*